# Systems and Software Producibility Collaboration and Experimental Environment (SPRUCE)

## Software and Systems Technology Conference
## April 20-23, 2009

Steven Drager[1], Rick Buskens[2], Patrick Lardieri[2], William McKeever[1]
[1]Air Force Research Laboratory, Rome, NY
[2]Lockheed Martin Advanced Technology Laboratories, Cherry Hill, NJ


PA Approval: #88ABW-2009-1126 dated 20 Mar 2009

# Outline

- **Problem**
- **Proposed Solution: SPRUCE**
  - **Overview**
  - **Architecture**
  - **Implementation**
  - **Challenge Problems**
  - **Timeline**
- **Summary**

# Problem

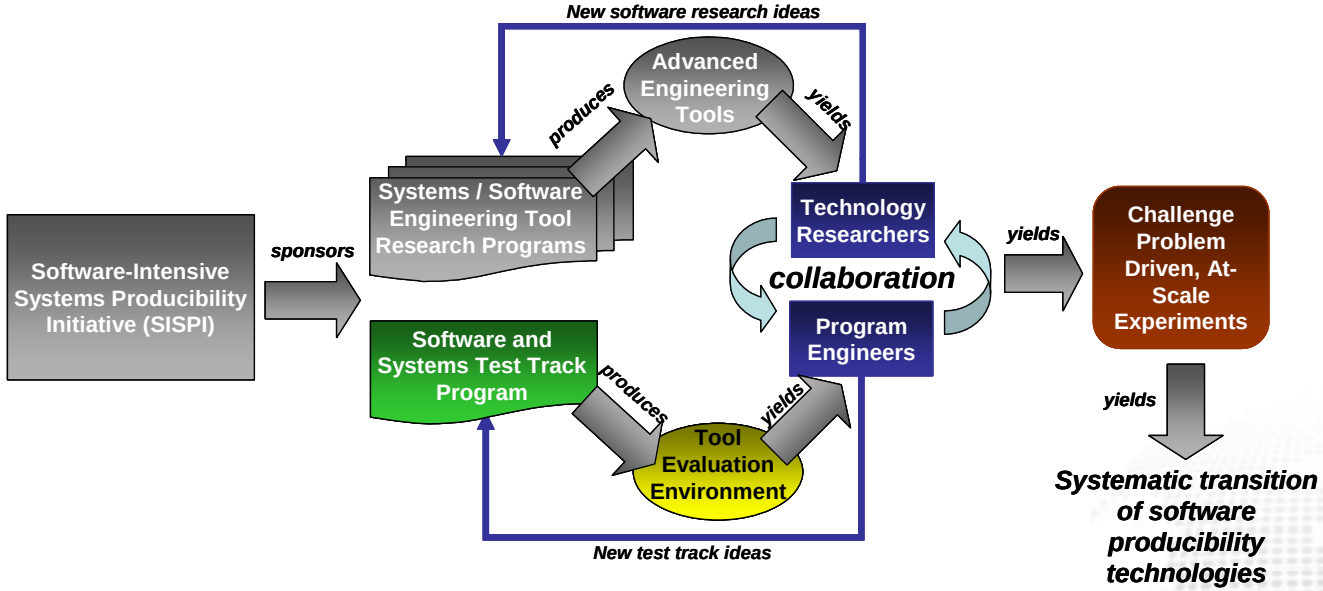
- **Software-intensive systems producibility for the DoD is inherently complex**
  - **Huge scale: $10^5$ requirements, $10^7$ lines of code**
  - **Huge number of component interactions**
  - **Increasing use of COTS and 3rd party software**
  - **Long (20-40) year product lifecycles**
  - **Stringent certification standards**
- **Difficult to introduce advanced technologies into DoD acquisition programs**
  - **Insufficient evidence to prove capabilities**
  - **Immature prototypes lack stability, features, support, tool chain integrations**
  - **Challenging to evaluate technology against "realistic" DoD problems**
  - **Must manage restrictions of classified, proprietary and ITAR information**
- **Root cause: *ad hoc collaboration among SISPI community members* ⇒ "valley of disappointment" for DoD programs**
  - **Unable to find SISPI technologies that meet needs and/or fail to adopt promising SISPI technologies**
  - **Software producibility problems encountered repeatedly across programs**
  - **Additional problem: "landing path" not typically DoD programs**

# Software-Intensive Systems Producibility Initiative

- Software is the prime enabler of complex weapons systems and command and control infrastructure
- Software is the least well understood and the most problematic element of large-scale systems
  - Little underlying science
  - Minimal engineering knowledge base
- *Software* and software project failures dominate causes of
  - *system* cost and schedule overruns
  - failures of *systems* to satisfy their requirements
  - Increasing numbers of costly and dangerous *system* failures
- Research to improve our ability to develop complex software-intensive systems is token, disjoint, and narrowly focused
  - Industry has no incentive to solve common problems
  - Academia sees no consistent funding stream

# SPRUCE Vision

- **Bring researchers together with developers and development artifacts in an open collaborative research and development environment to 'test drive' (demonstrate, evaluate, and document) the ability of novel tools, methods, techniques, and run-time technologies to yield more predictable production of software intensive systems**

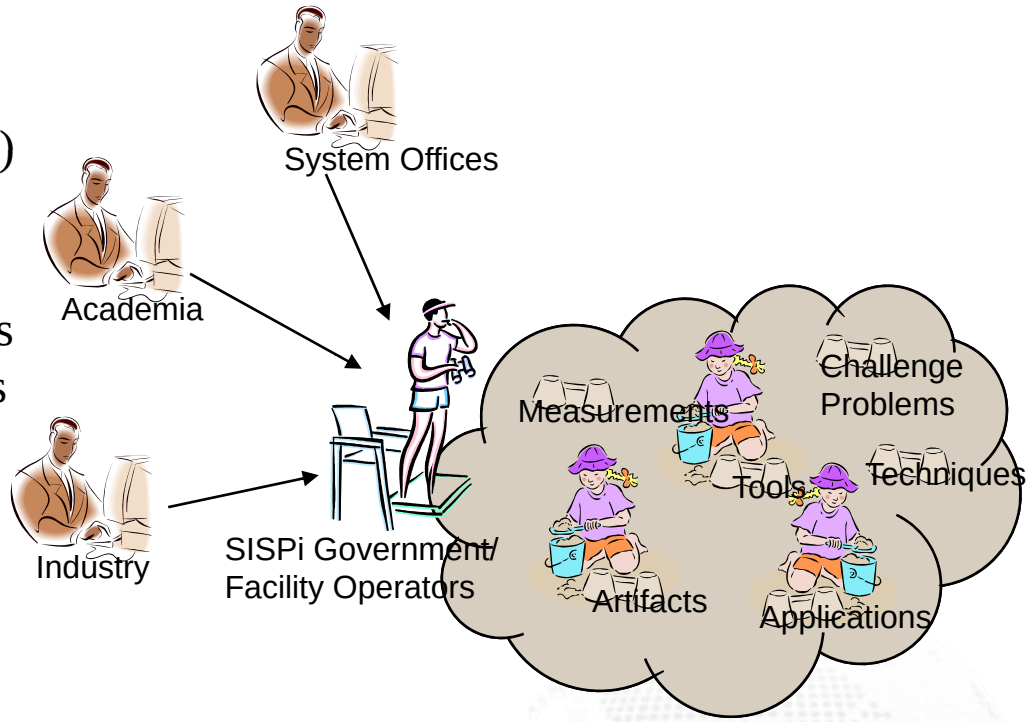



- **Benefit: challenge problem-driven, at-scale experiments can be defined & conducted to prove/disprove suitability of particular software producibility technologies**
  - **Increasingly successful and more systematic process for transition across the software producibility spectrum**
  - **Well connected, self-sustaining community of participants**

# Systems & Software PRodUcibility Collaboration & Experimentation Environment (SPRUCE)

## Approach:

- Provide access to representative artifacts
- Provide tools (e.g., simulation, emulation) to realistically test techniques/technologies
- Provide resources to support realistic tests and experiments on challenging problems
- Assess tool utility and document results and benefits to DoD acquisition community




## Challenges:

- Evaluating technology in an unbiased manner against "realistic" DoD problems
- Dealing with restrictions of classified, proprietary, and ITAR information
- Protecting IP
- Getting people actively participate and collaborate

**Payoff:** Rapid development of more robust software for DOD users through:

- Rapid identification of flaws
- Rapid transition of proprietary knowledge and processes
- Rapid, intelligent creation of collaboration teams

# SPRUCE

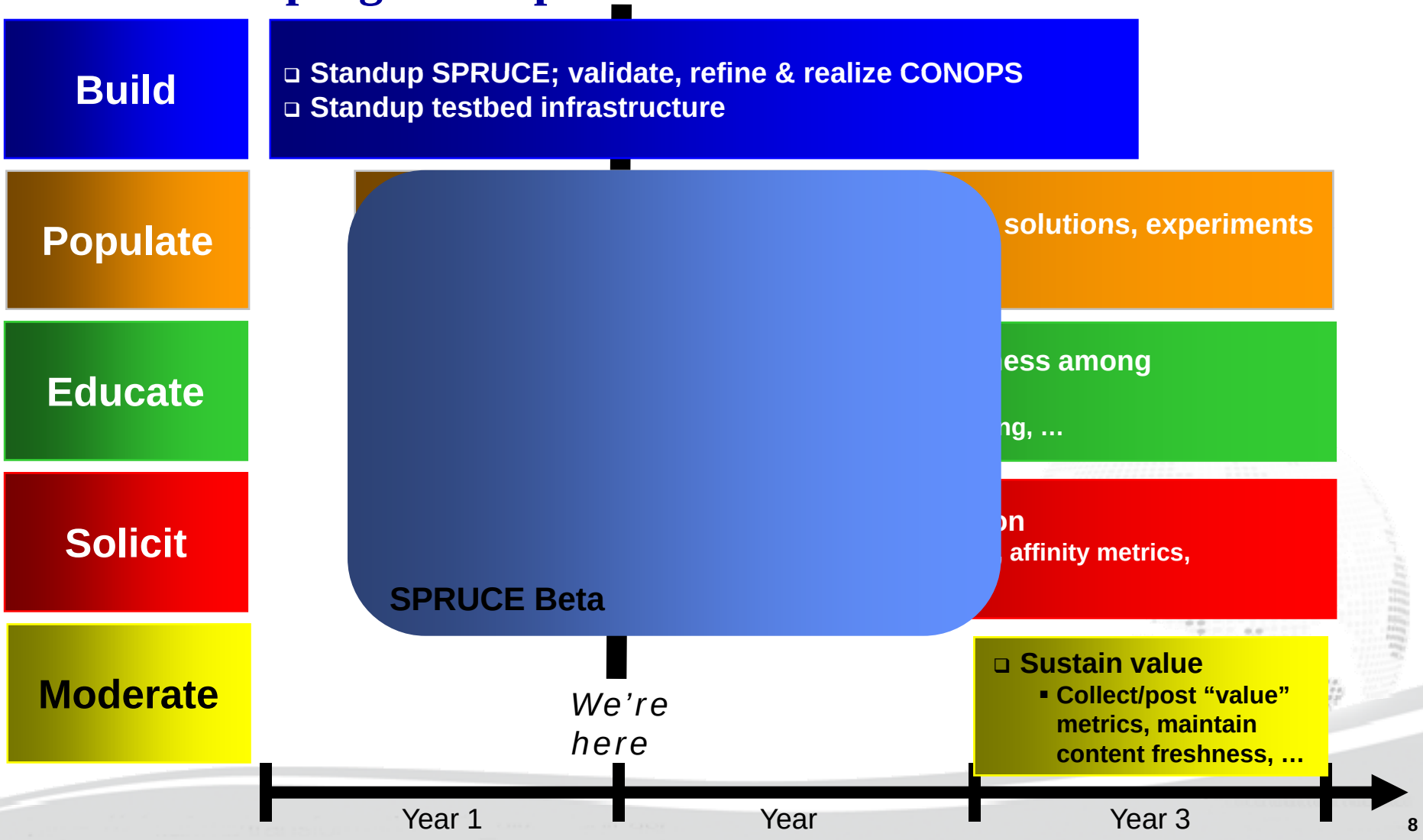
- **Systems & Software PRodUcibility Collaboration & Experimentation Environment ($S^2PRUCE^2$ ⇒ SPRUCE)**

- **An open, collaborative research, and development environment to demonstrate, evaluate, and document the ability of novel tools, methods, techniques, and run-time technologies to yield affordable and more predictable production of software intensive systems**
  - **Brings together researchers, developers and domain experts from different communities to de-fragment the knowledge necessary to achieve SISPI research, development and technology transition**

- **Capabilities**
  1. ***Experimentation* on novel tools, techniques and methods**
  2. **Define *challenge problems* and *associated artifacts* to demonstrate the challenge problems**
  3. **Propose *candidate solutions* to challenge problems – i.e., technologies/tools/methods that address some or all parts of a challenge problem**
  4. **Define and conduct realistic *experiments* to assess candidate solution utility and document results and benefits to the DoD acquisition community**
  5. ***Collaborate* on all of the above**

# At A Glance

- **36-month program: April 2008 – March 2011**



# Build Plan

## SPRUCE CONOPS & Design

**Functional Architecture**

- Candidate Solution, Challenge Problem Management, Challenge Problem, Experiment, Collaboration, Account, & infrastructure Management
- Experimental & Operations Infrastructure

**Operational Scenarios**

- Identify & Evaluate Challenge Problems
- Propose Candidate Solutions
- Design & Conduct Experiments
- Benchmark, Validate, & Adopt Solutions
- Collaborate on Challenge Problems, Solutions, & Experiments

**User Roles**

- Challenge Problem Provider
- Candidate Solution Provider
- Experimenter
- Collaborator
- Administrator

## SPRUCE Deployment Architecture

**Physical Infrastructure**

**Consolidated Interface**

- Web-based portal interface to provide a consolidated view of SPRUCE information, tools, & events, & user activity

**Taxonomy**

- Reflects the functional architecture components
- Accommodates & encourages ad hoc collaboration

**Data & Access Control Model**

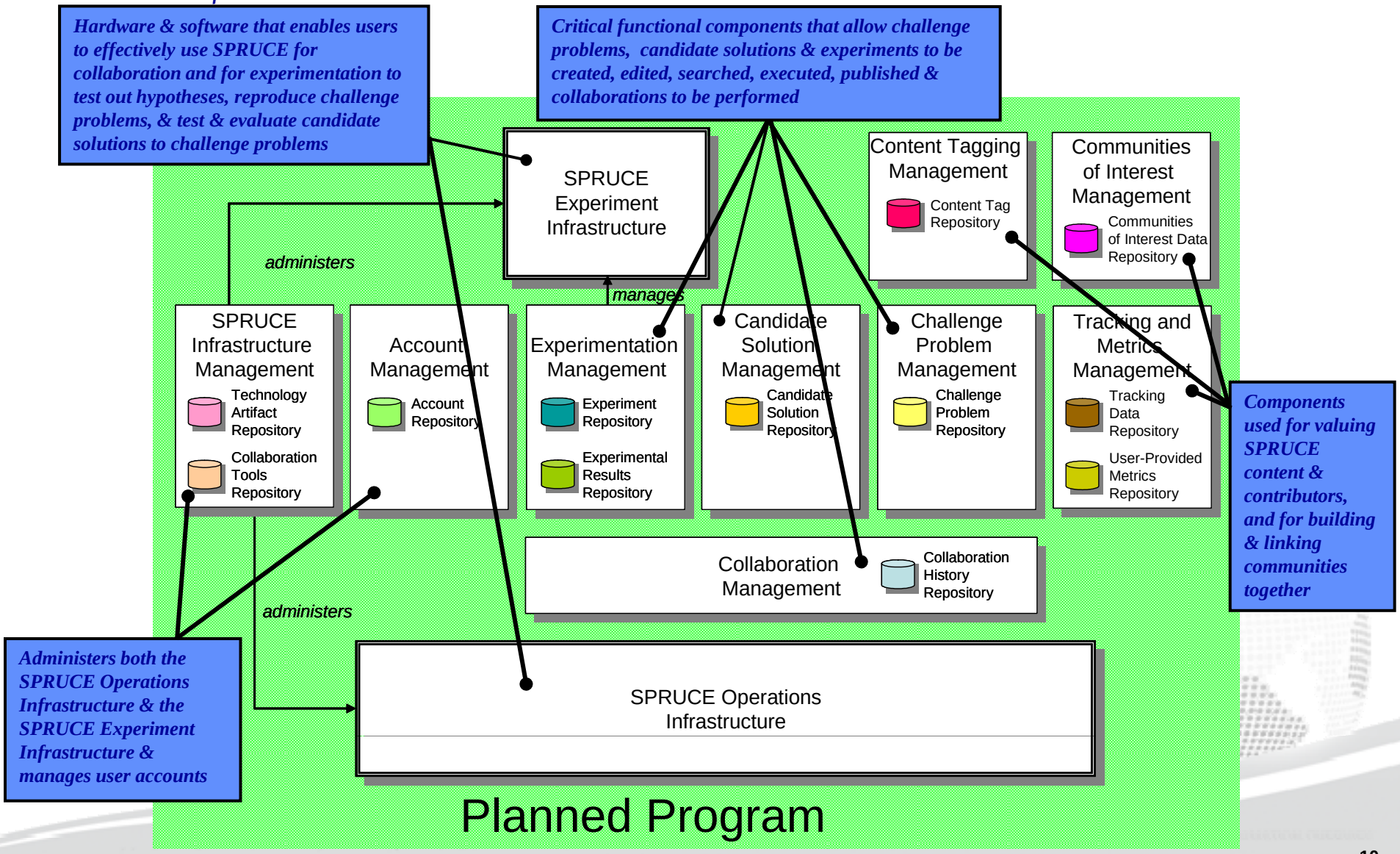
- Data describing users, challenge problems, candidate solutions
- e.g. Problem Type, Domain, Characteristics
- Permissions to access areas of the Portal

**Workflows**

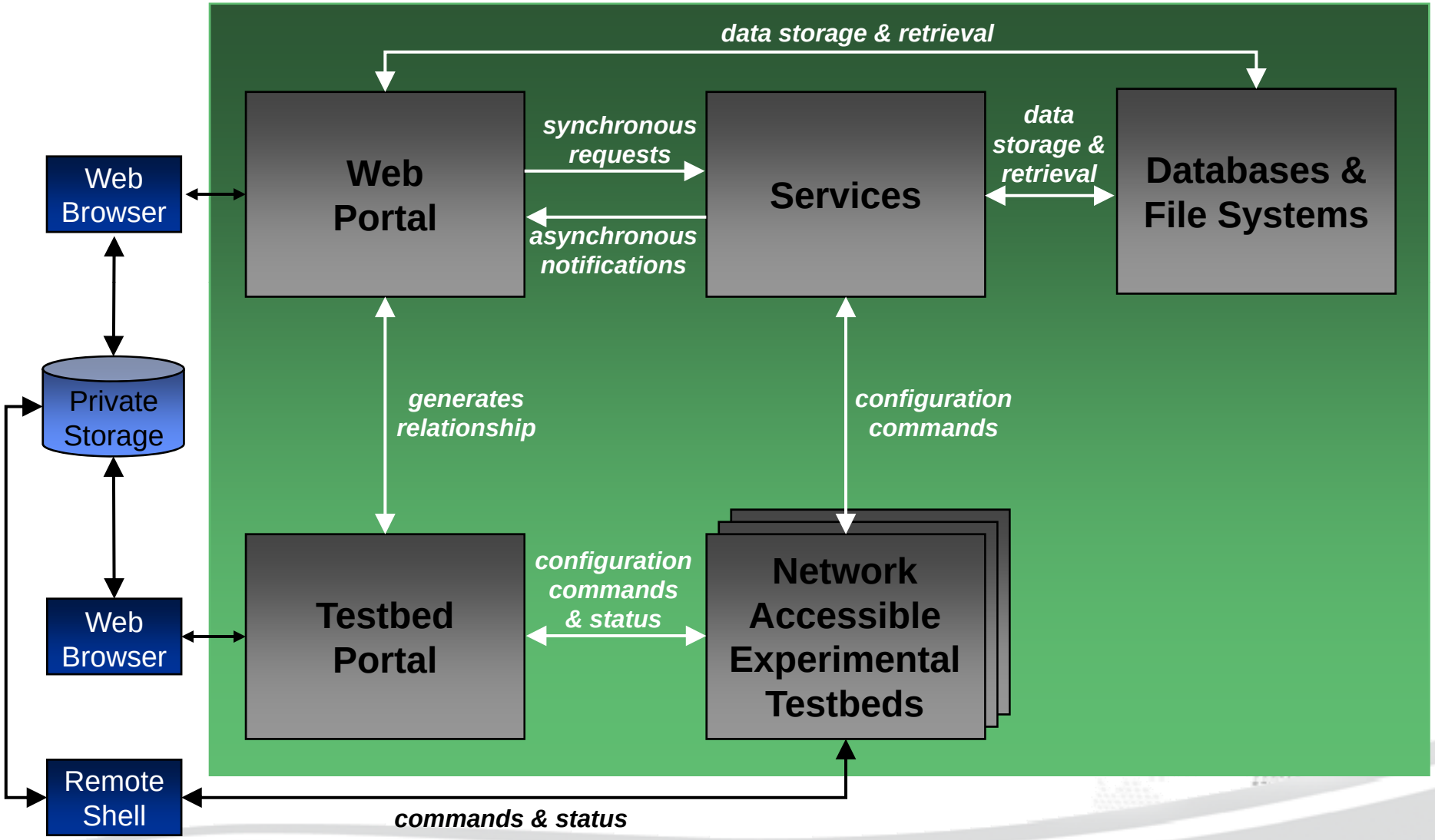
- Realize the processes to facilitate Operational Scenarios
- Modularized to flexibly accommodate alternate & iterative flows

- **SPRUCE participant activities & interactions are envisioned to be parallel, iterative, & complex. The proposed SPRUCE environments will seek an organizing construct to accommodate variations of activity sequences & collaborations by providing modular workflows & ad hoc collaboration spaces**

# Functional Architecture

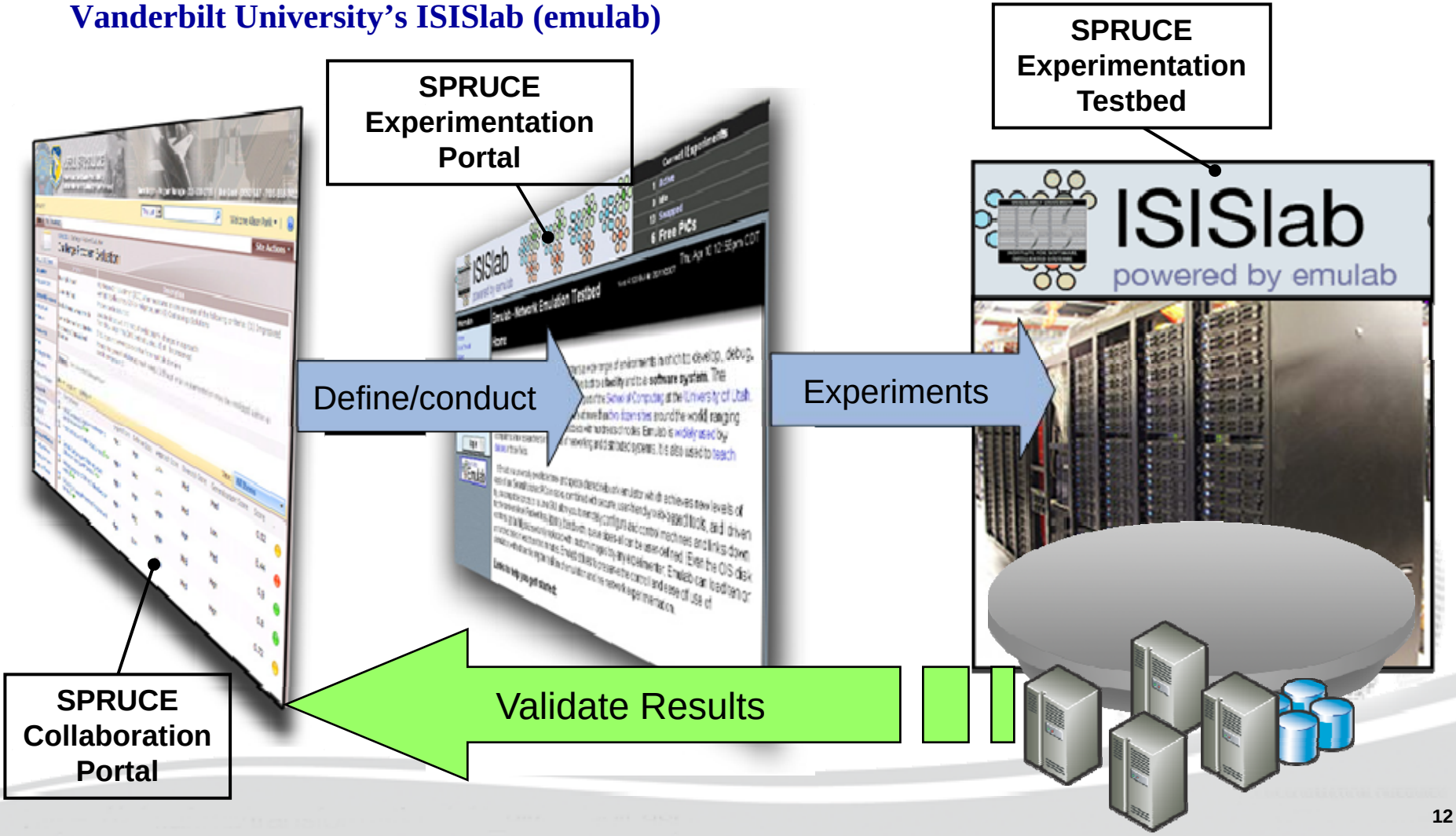


# Implementation Architecture

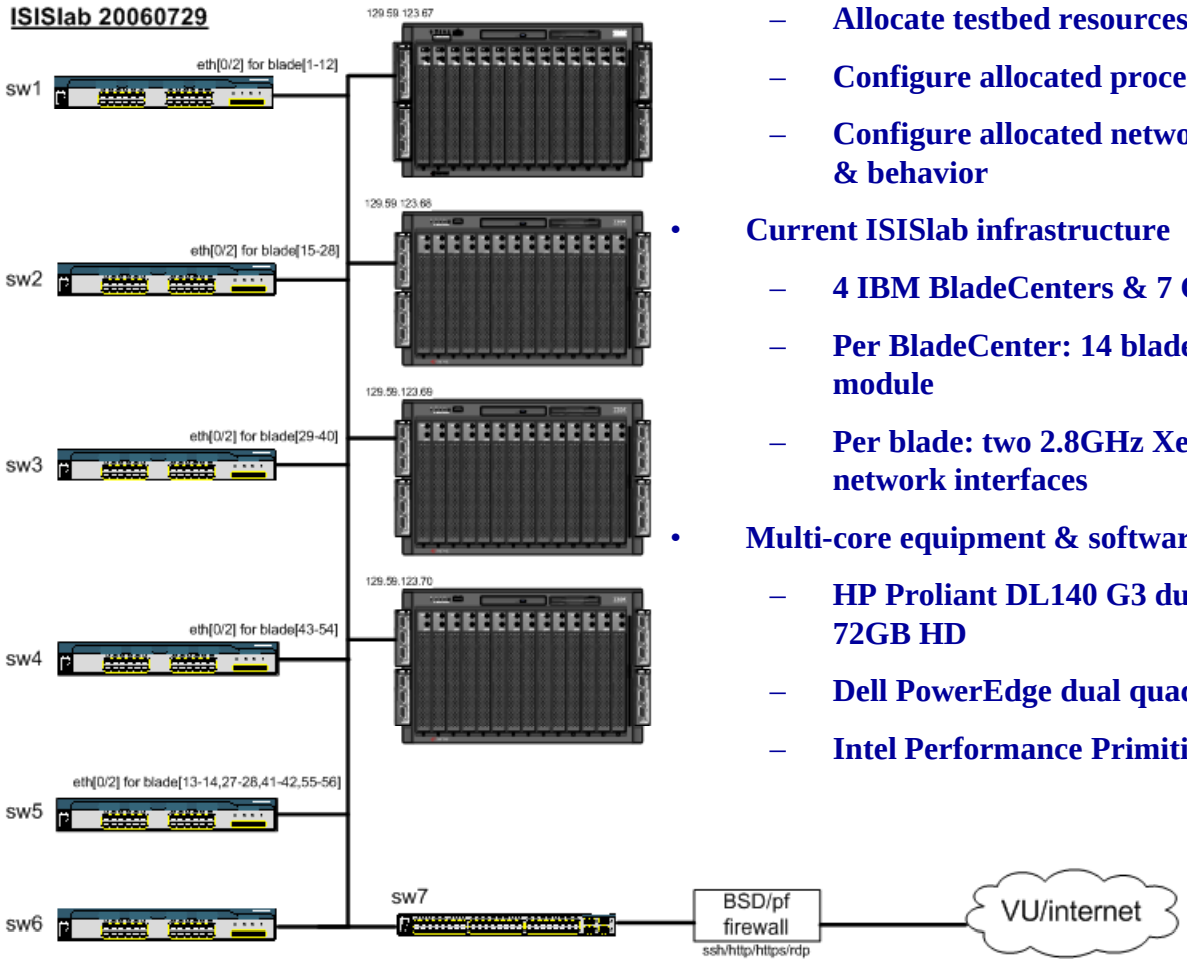


# Implementation Platform Overview

- **SPRUCE platform =**
  **Customized collaboration portal +**
  **Vanderbilt University's ISISlab (emulab)**



# Experimentation Testbed



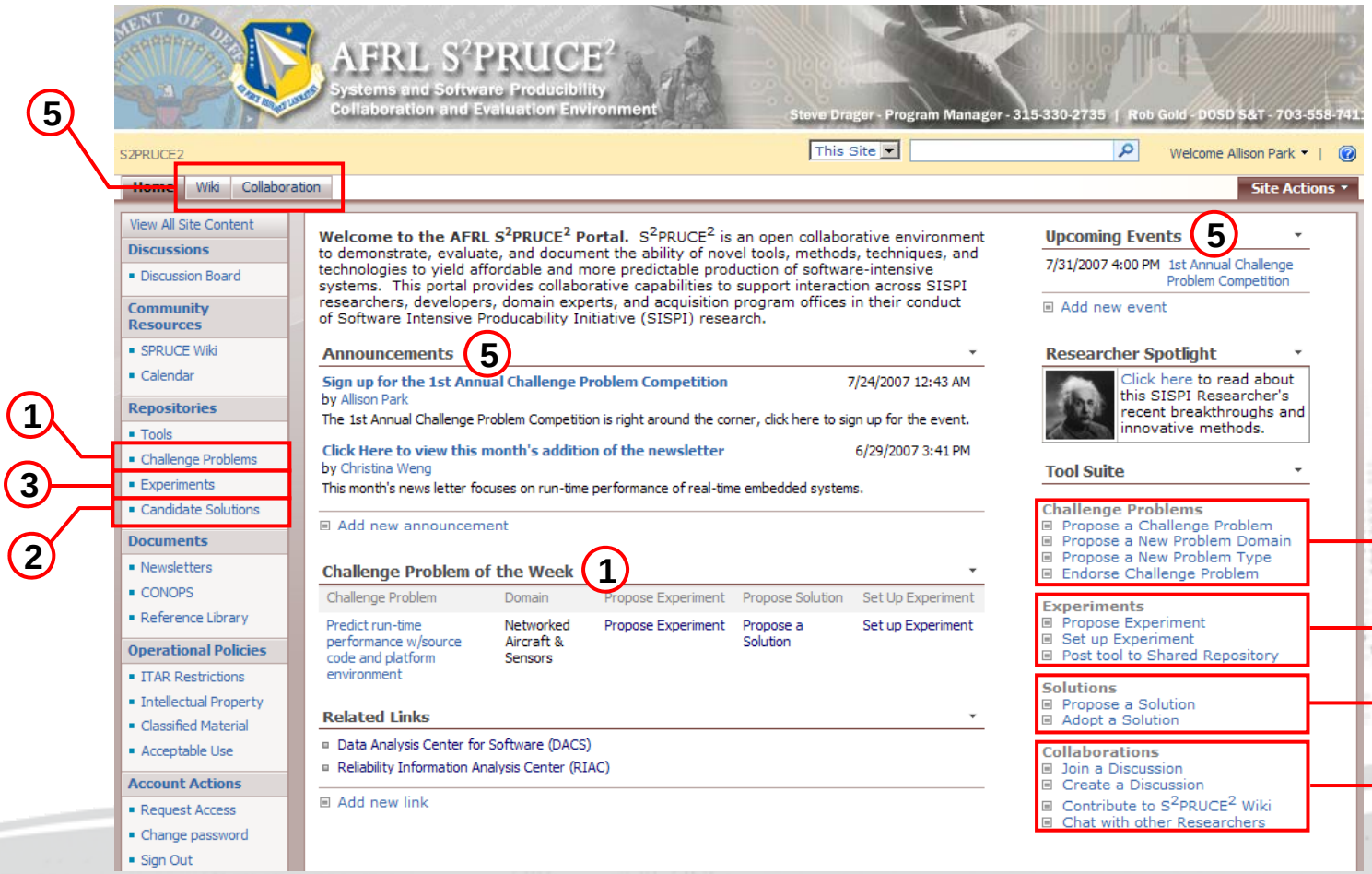

- **Emulab infrastructure (www.emulab.net)**
  - **Allocate testbed resources for research & experimentation**
  - **Configure allocated processors to run any supported operating system**
  - **Configure allocated network resources to different network topologies & behavior**
- **Current ISISlab infrastructure**
  - **4 IBM BladeCenters & 7 Cisco switches**
  - **Per BladeCenter: 14 blades, 4 Gbps network I/O modules; 1 mgmt module**
  - **Per blade: two 2.8GHz Xeon CPUs, 1GB RAM, 4GB HD, 4Gbps network interfaces**
- **Multi-core equipment & software**
  - **HP Proliant DL140 G3 dual quad-core Intel Xeon E5320, 4GB RAM, 72GB HD**
  - **Dell PowerEdge dual quad-core Xeon, 32GB RAM, 2TB HD**
  - **Intel Performance Primitives 5.1**

# Portal Interface

AFRL — THE AIR FORCE RESEARCH LABORATORY — LEAD | DISCOVER | DEVELOP | DELIVER

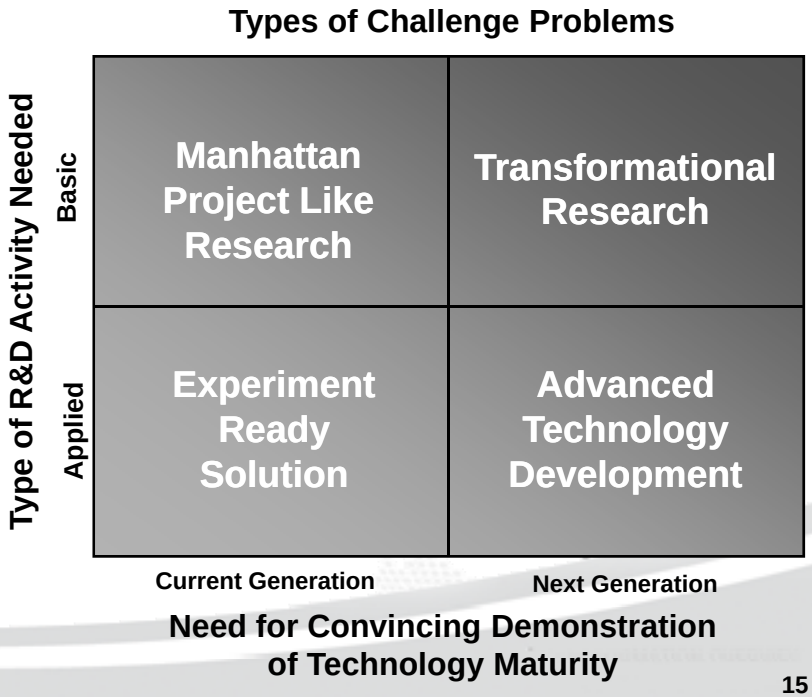
*Information Directorate*

| Identify & Evaluate Challenge Problems (1) | Propose Candidate Solutions (2) | Design & Conduct Experiments (3) | Benchmark, Validate, & Adopt Solutions (4) | Collaborate on Challenge Problems, Solutions, & Experiments (5) |
|---|---|---|---|---|



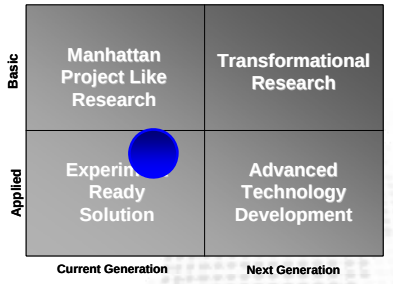
# Challenge Problems in SPRUCE

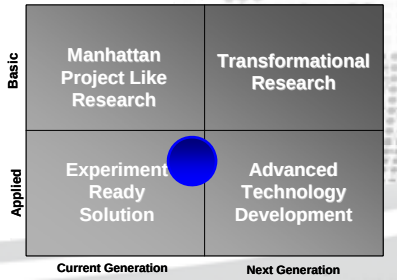
- A description of a software producibility deficiency that negatively impacts quality, capability, cost or schedule of software intensive DoD development programs.
  - e.g., assuring no race conditions in multi-threaded C++ or Java source code
- Organized into broad areas that contain one or more specific challenge problem instances
  - e.g., the problem described above could be a specific instance in the broad area of assuring non-functional run-time properties
- Types of challenge problems suitable for SPRUCE:
  - Type of R&D activity needed
    - Basic – focuses on fundamental scientific or technology gap
    - Applied – focuses on incremental capability gap or performance improvement beyond state of practice
  - Transition timeframe needed
    - Current generation programs – DoD acquisitions over next 1-5 years
    - Next generation programs – DoD acquisitions over next 5-10 years



# Sample Challenge Problem Instances

- **Predicting Computer System Capacity Needs**
  - **A modern computing architecture consists processors, memory, bus controllers, bridges, arbiters, DMA controllers and other components. The total behavior of the system is the interdependency of the processor configuration, the software load & the message load on the system. There is a need for a predictive model that incorporates these elements into an environment suitable for trade studies & processor system specification.**
- **Seed Challenge Problem Instances**
  - **Predicting Performance of Avionics Mission Systems Using a Logical Task Network & Computing Plant Description**
    - **Sponsored by Lockheed Martin Aeronautics**
    - **Seeking Experiment Ready Solution for F-35 & F-22 programs**
  - **Optimization Techniques for Deployment of Avionics Mission Systems onto a Specific Computing Plant**
    - **Sponsored by Lockheed Martin Aeronautics**
    - **Seeking Experiment Ready Solution for F-35 & F-22 programs**

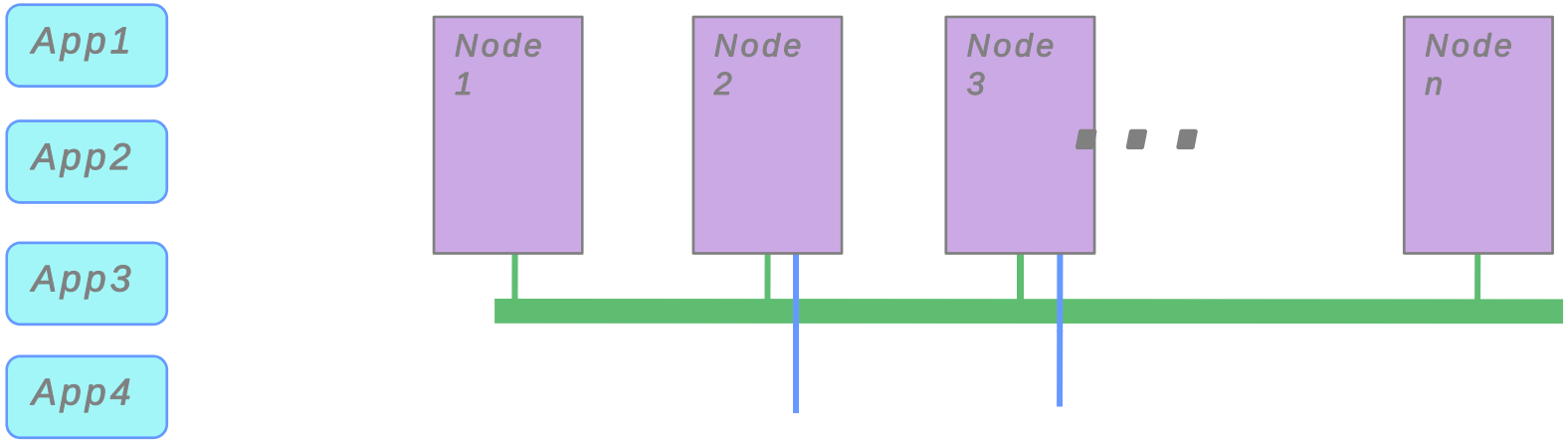


# Sample Challenge Problem: From Lockheed Martin Aeronautics

- **Representative avionics domain artifacts**
  - **Though not specific to any particular aircraft program, the types and volumes of data are accurate**
  - **Multiple vendors and development teams work independently – integration happens several times a year**
  - **Presented as a Microsoft Excel workbook**
- **Targets for optimization**
  - **CPU count: reducing the number of CPU's required directly impacts weight, cooling, power on aircraft**
  - **Network bandwidth: minimizing network traffic through smart co-location of software applications**
  - **Internal bus (e.g., PCI): minimizing impacts of I/O blocking CPU cache miss servicing through smart scheduling of I/O, cache miss predictions per application**

# Sample Challenge Problem: Notional Computing Architecture




- Nodes have defined resource levels, network connections, etc.
- Applications have defined resource requirements (CPU, I/O)
- Most applications are node-neutral – i.e., can be run on any node with sufficient resources (though a few require specific hardware)
- Architecture short-circuits communications between applications on the same node

# Sample Challenge Problem: Artifact Representation

- **Generic System I/O Signal Set**
  - **Characteristics of every message in system**
  - **Sending and receiving applications (can be multiple)**
  - **I/O port on sending and receiving sides**
  - **Type: startup, periodic, aperiodic, etc.**
  - **Size and frequency of transmission**
- **Generic Processor Loads**
  - **Nodes are all assumed to be identical resource levels**
  - **Applications defined in terms of induced CPU load per execution frequency**
  - **Assumed rate monotonic scheduling at a base rate of 50Hz (20 millisecond frame)**

| | | Rate | | | |
|---|---|---|---|---|---|
| **Processor ID** | **Application** | **0.195313** | **0.390625** | **0.78125** | **1.5625** |
| 25 | App1 | 0.00% | 0.05% | 0.04% | 0.33% |
| 27 | App2 | 0.00% | 0.05% | 0.04% | 0.33% |
| 22 | App3 | 0.00% | 0.46% | 0.41% | **60.50%** |

*App3 on Processor 22 has a module which executes 1.5625 times per second (640 millisecond frame) and represents a CPU load of 60.50%, so App3's 1.5625Hz component executes 60.50% X 640 = 387 milliseconds per release*

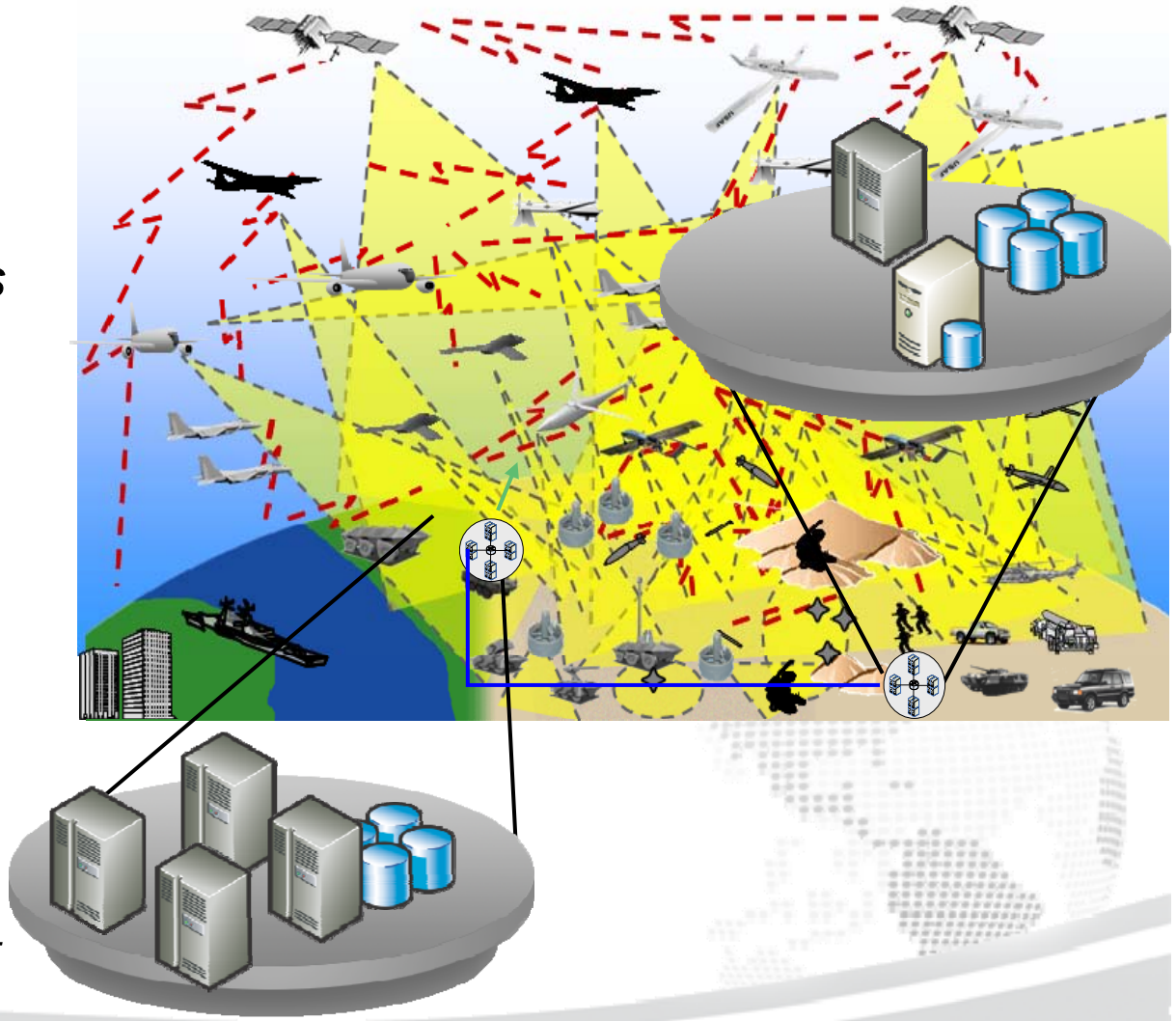
# Candidate Solution

- **Describes a potential solution to a challenge problem**
  - **May or may not include a realization of the solution (if a tool or technology)**
- **Solution technologies in SPRUCE consist of:**
  - *Name*
  - *Description / capability* **- high-level view of how the solution works**
  - *Type of solution* **– e.g., process, tool, technology**
  - *Domain of applicability* **– what's in scope, out of scope**
  - *Licensing conditions* **– terms for licensing the technology, if any**
  - *Development platform* **– important if tool/technology is to be extended by other SPRUCE users**
  - *Run-time platform* **– helps determine requirements for tool usage in actual program environments**
  - *Associated challenge problems* **– which challenge problem(s) the solution is intended to address**
  - *Associated solution technologies* **– related solution technologies**

# Sample Candidate Solution Match: GUTS from Vanderbilt University

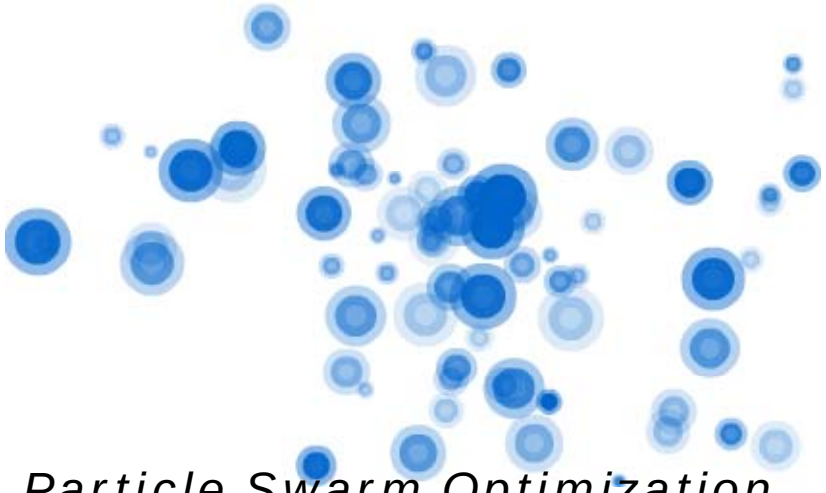
*GUTS: System Execution Modeling Technologies for Large-scale Net-centric DoD Systems*

1. *Existing 6.1 basic research effort funded by AFRL, but not as part the SPRUCE effort.*
2. *Goal: Build tools to identify system deployment topologies that meet performance requirements*



# Sample Candidate Solution: Solution Approach

***Existing techniques weren't designed to handle network bandwidth minimization or real-time scheduling constraints.***



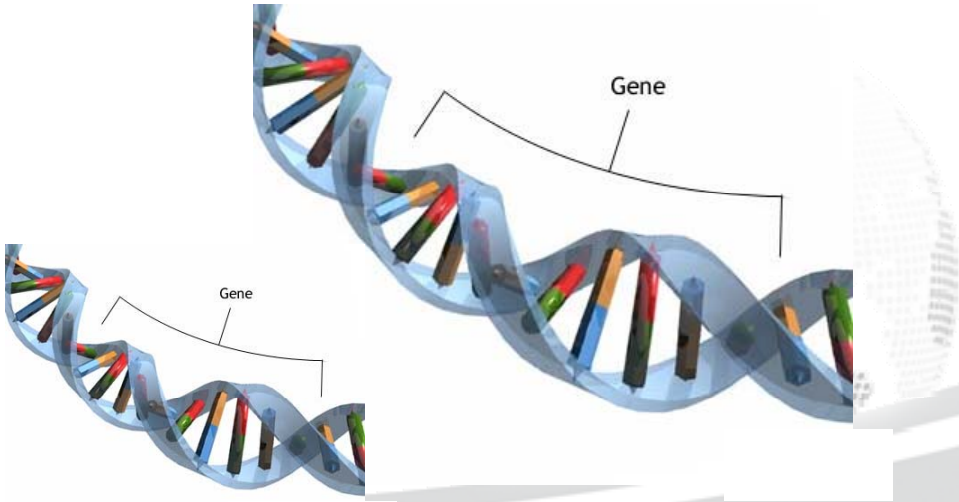
Particle Swarm Optimization (PSO)

## Solution Approach:

Modified bin-packing algorithms to support scheduling constraints using Liu & Layland bound

Combined randomized bin-packing with particle swarm and genetic optimization algorithms to search for bandwidth minimizing deployment topologies

Connectors to maintain consistency between deployment models

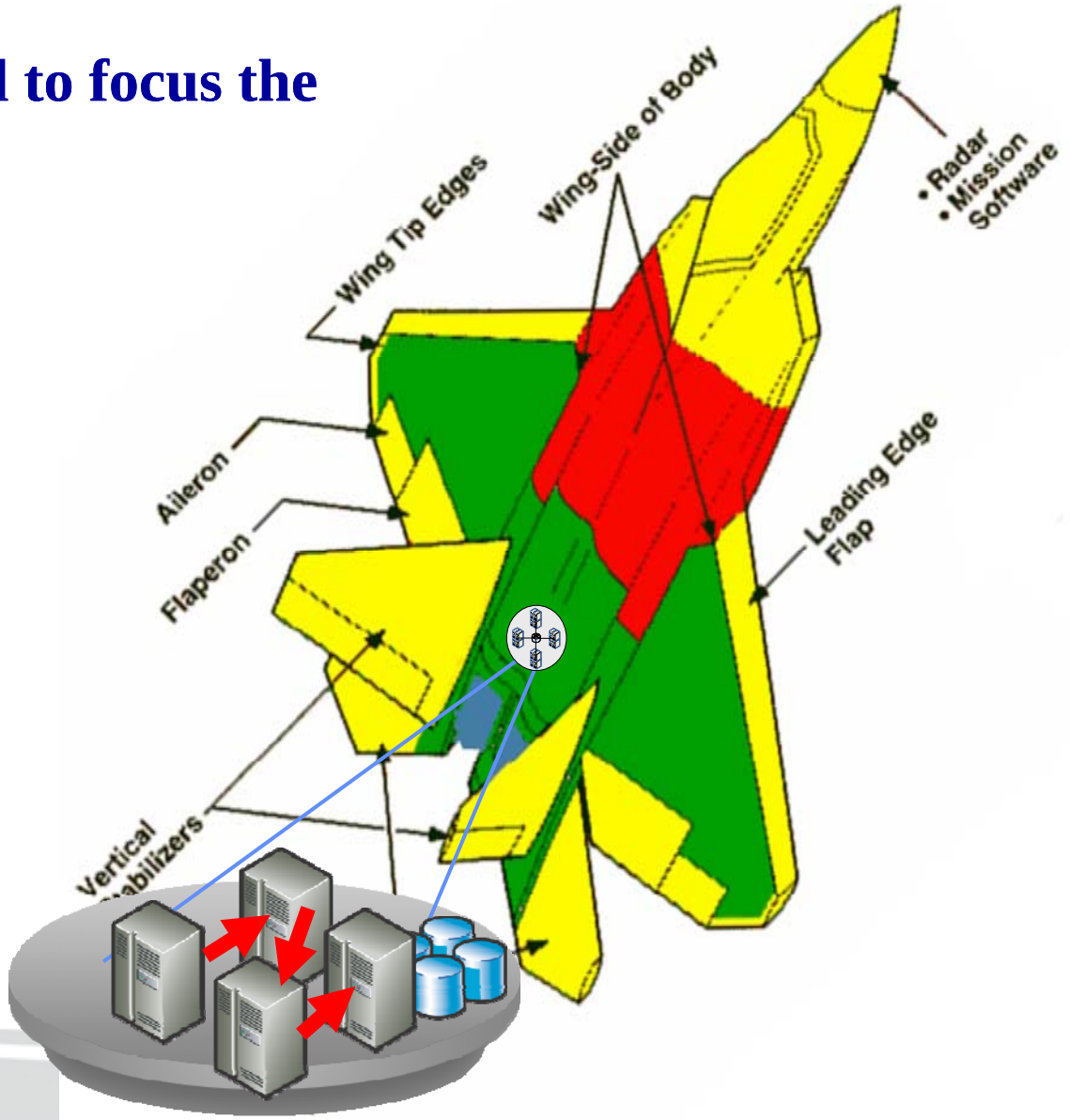



Genetic Optimization

# Sample Candidate Solution: SPRUCE's value

## SPRUCE portal helped to focus the research efforts:

1. *Challenge problem provided realistic data to test the deployment techniques on*
   - *Previously, relied on ad hoc data to approximate large-scale problems*
2. *Challenge problem definitions helped to plan research extensions that will have most impact*
3. *Papers much easier to write, motivate, & provide empirical results*

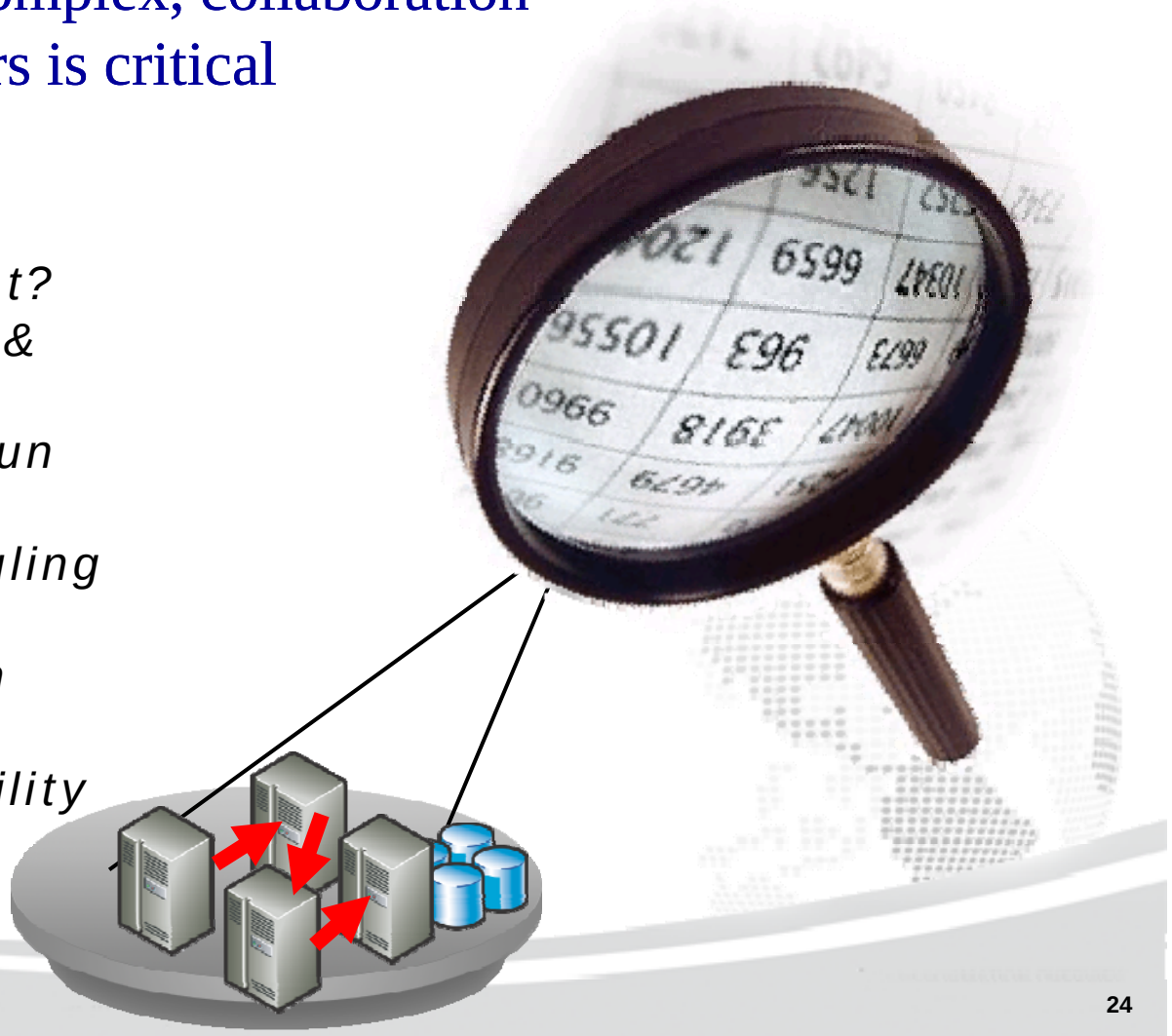


# Sample Collaboration: Helping Problem Understanding

Since the challenge problem datasets are very large & the challenges complex, collaboration with problem contributors is critical

*Examples from this set:*

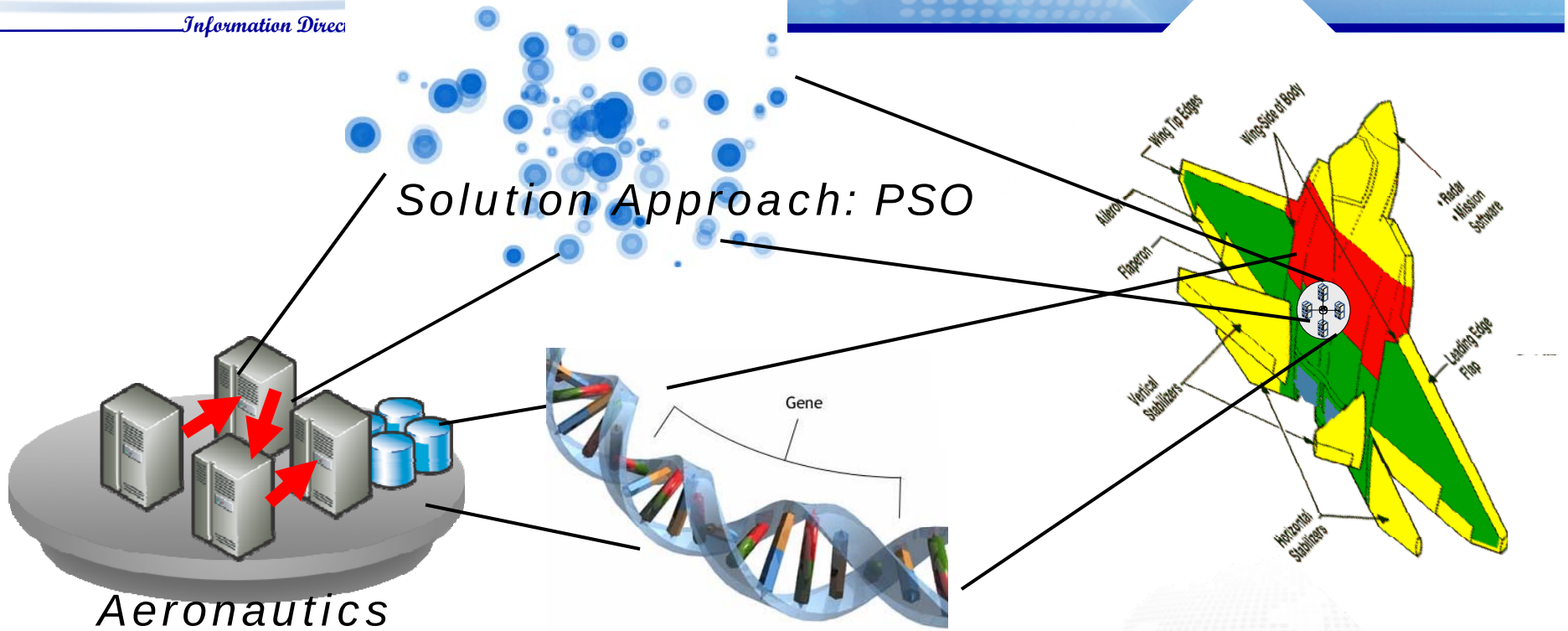
1. *Are priorities present? Are they predefined & static? Dynamic?*
2. *Can multiple tasks run at the same time?*
3. *What type of scheduling is appropriate?*
4. *How is task duration calculated?*
5. *What are the scalability requirements?*



# Experiments

- **Used to evaluate the suitability/benefits of a particular solution technology applied to a specific challenge problem**
  - **Associated with challenge problems *OR* solution technologies**
  - **Developed by *any* SPRUCE user**
- **Experiments in SPRUCE consist of:**
  - *Objectives* **– summary of specific objectives of the experiment**
  - *Associated challenge problems* **– which challenge problems the experiment is intended to evaluate against**
  - *Associated solution technologies* **– which solution technologies are suitable for running the experiments**
    - **Experiments may be used for benchmarking**
  - *Experimental configuration & artifacts* **– so that the experiment can be reliably reproduced**
  - *Benchmarks* **– baseline to measure results against**
  - *Expected results* **– expectations of results to be produced**
- **To demonstrate suitability of a particular solution technology to a specific challenge problem**
  - **Must store results of executed experiments in SPRUCE**
  - **Must provide details on how to analyze/interpret experimental results**

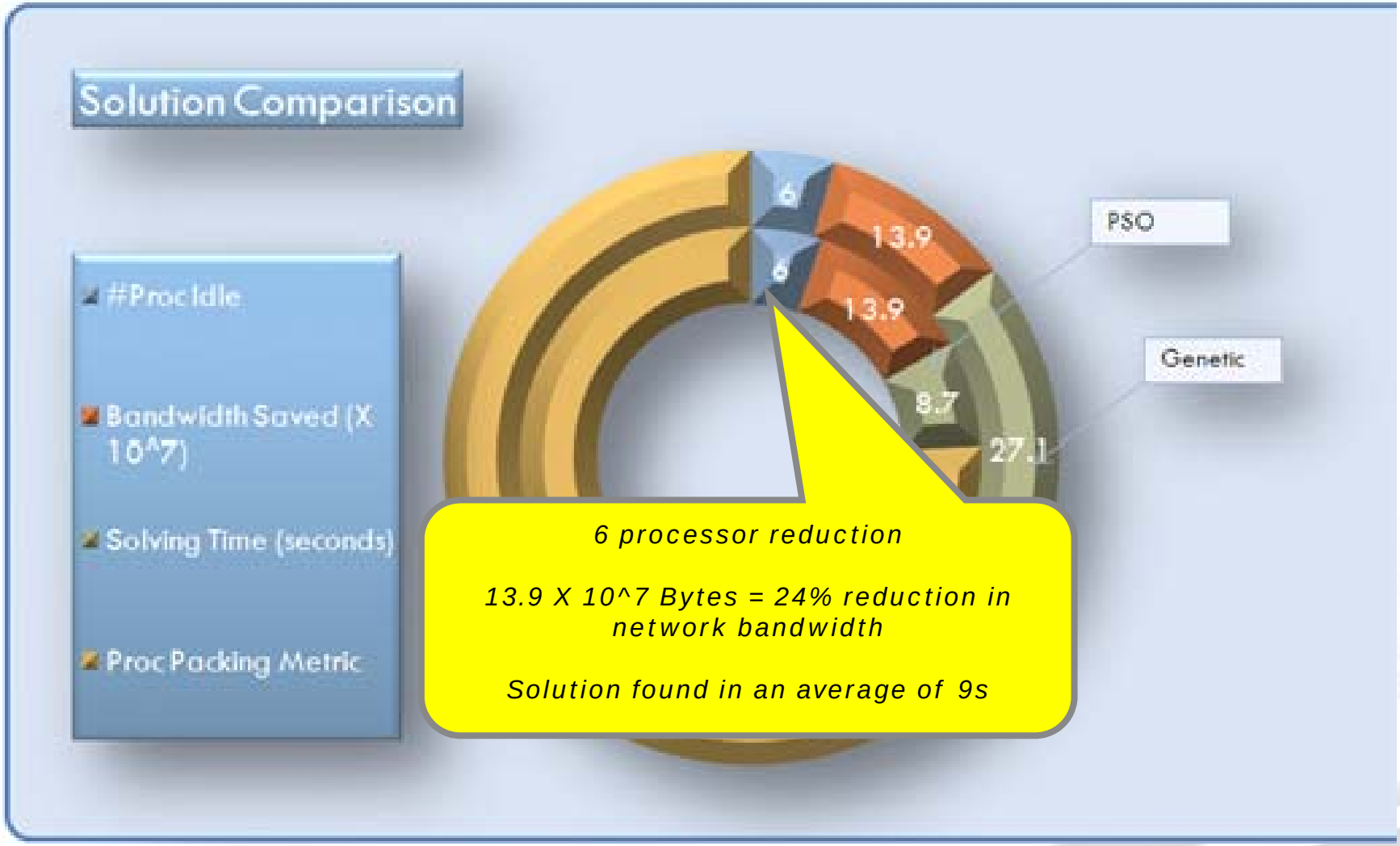
# Sample Setup of SPRUCE Experiment




***Avionics Experiment Setup:***

- ***Mission avionics system data from Lockheed Martin comprising 41 processors, 41 software components, ~150 real-time tasks, & ~14,000 component interactions***
- *Attempted to minimize total processors used*
- *Attempted to minimize total network bandwidth consumed*

# Sample Results on from Example



# Focus on Challenge Problem: Foundation for Success Criteria

- **Challenge problems are the "killer content" that transforms SPRUCE from a web portal into a virtual community by:**
  1. **Providing a conceptual foundation that attracts the primary groups – SPOs, contractors, researchers, & commercial vendors – in successful SISPI research collaborations**
  2. **Providing real-world context, constraints & metrics to evaluate SISPI research progress**
  3. **Providing well-defined transition & commercial tool opportunities for mature SISPI technology with proven results**

**Example: ATL QoS website**

- Dedicated to the challenge of building distributed real-time embedded (DRE) systems from open QoS-enabled middleware, operating systems & networks
- Has attracted a participant community whose collaborations have driven middleware technology development
  - **SPOs: DDG-1000, Aegis**
  - **Contractors: Boeing, Raytheon, Lockheed Martin, etc.**
  - **Universities: Vanderbilt, Wash U, CMU, Drexel, …**
  - **Commercial vendors: OIS, ZeroC, PrismTech, etc.**

**Example: Boeing Open Experimental Platform (OEP)**

- DARPA PCES program
- Provided dozens of avionics system scenarios & software artifacts
- Drove evaluation of program composition technologies
- Ultimately established empirical evidence that enabled transition of ESCHER modeling tools onto FCS

**Example: CMU's Fluid Java code analysis tools**

- Lockheed Martin's Software Technology Initiative (STI) matched CMU's mature Fluid Java code analysis technology to the U.S. Navy's Aegis Open Architecture program & DDG-1000 programs
- Helped to establish business case for SureLogic startup & submission of congressional plus-up proposal between LM MS2-Moorestown, PMS 500, & SureLogic to apply SureLogic code analyers to DDG-1000 R5 software QA

# Status

- **Kickoff – April 2008**
- **Spiral 0 – June 2008**
  - **Out-of-the-box (OOTB) capabilities**
    - **Create collaboration spaces and repositories associated with challenge problems, candidate solutions and experiments**
    - **Basic support to edit, delete, search for information**
    - **Populate SPRUCE with initial challenge problem**
- **Spiral 1 – August 2008**
  - **OOTB customization of fundamental capabilities – *SPRUCE beta***
    - **Support linking challenge problems, candidate solutions and experiments**
    - **Invite collaborations**
    - **View collaboration histories**
    - **Full support to edit, delete, search for information**
    - **Populate SPRUCE with initial challenge problem**

# Future Milestones

- **Spiral 2 – June 2009**
  - **SPRUCE populated with four additional challenge problems (Dec 2008)**
  - **Customized collaboration and workflow enhancements**
    - **Tighter integration between SPRUCE operations portal and experiment infrastructure**
- **Spiral 3 – June 2010**
  - **Usage patterns, benchmarking and trend intelligence**
    - **Track usage metrics**
    - **Track and report affinities between researchers, problems, solutions and tools**

# *Imagine A Future, Where…*

- **As a Program Engineer toiling away, putting out day-to-day fires**
  - **…you can quickly & easily discover & reach out to a broad community of people who can relate to your technical problems**
  - **…you can quickly & easily learn about technologies that may help solve your problem**
  - **…you can readily engage a community of experts to solve your problem**
  - **As icing on the cake, DoD program managers are watching this activity, to solicit your input in their next dream project**
- **As a SISPI researcher building exciting technologies & tools**
  - **…you had a repository of real-world problems to work with**
  - **…you can quickly & easily engage an active practitioner community**
  - **…you had a means to effectively demonstrate how your technologies & tools could solve real-world problems**
- **As a DoD Program Manager**
  - **…you had access to a continuously evolving virtual community of people, problems & technologies from both constituents above**
  - **…and that these constituents can contribute critical insights, data & artifacts to get your program justified, started & transitioned!**

# Summary

- **SPRUCE: an open, collaborative research, and development environment to demonstrate, evaluate, and document the ability of novel tools, methods, techniques, and run-time technologies to yield affordable and more predictable production of software intensive systems**
  - **Brings researchers together with developers and development artifacts to 'test drive' emerging technologies and techniques**
  - **Includes testbed resources and tools specialized to particular operational domains (e.g., avionics)**
- **Benefit: challenge problem-driven, at-scale experiments can be defined and conducted to evaluate particular software producibility technologies**
  - **Increasingly successful *transition* across the software producibility spectrum and across operational domains**
  - **Well-connected, self-sustaining *community* of participants**
- **For SPRUCE to be successful, *WE NEED YOUR PARTICIPATION***

# POC Information

**Steven Drager**
**Air Force Research Laboratory/RITB**
**525 Brooks Rd.**
**Rome, NY 13441**
**TEL: +1.315.330.2735**
**Email: Steven.Drager@rl.af.mil**

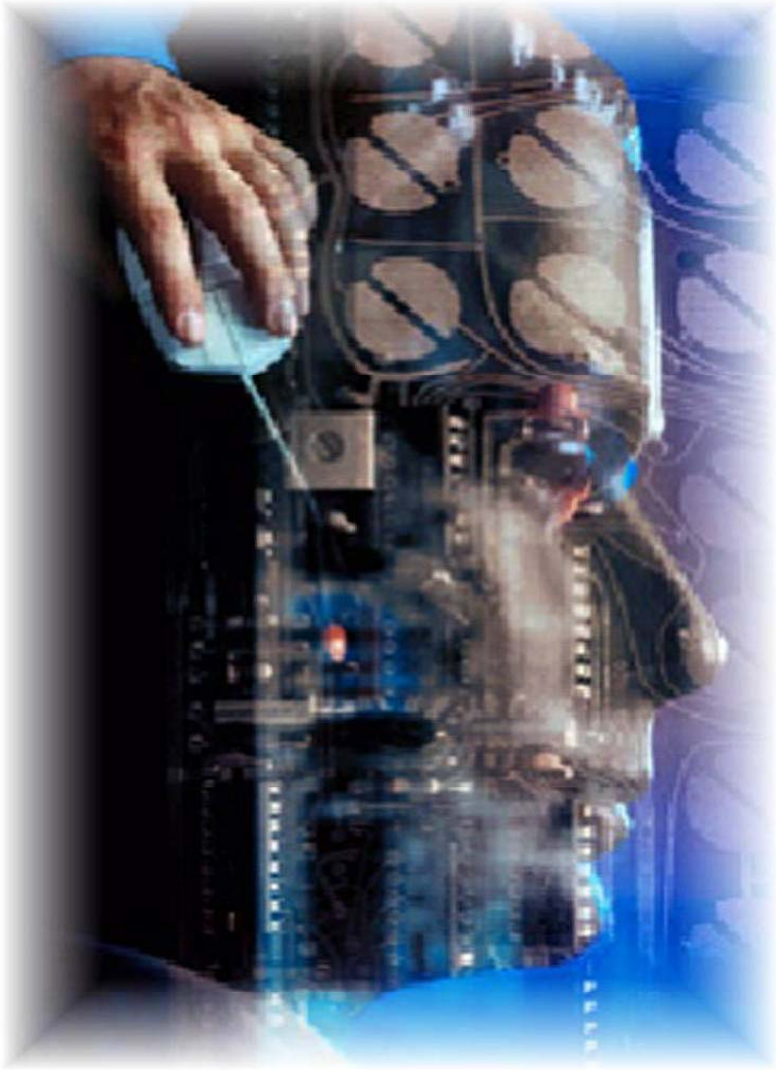
**William McKeever**
**Air Force Research Laboratory/RITB**
**525 Brooks Rd.**
**Rome, NY 13441**
**TEL: +1.315.330.2897**
**Email: William.McKeever@rl.af.mil**

**Dr. Richard Buskens**
**Lockheed Martin**
**Advanced Technology Laboratories**
**3 Executive Campus, 6th Floor**
**Cherry Hill, NJ 08002**
**TEL: +1.856.792.9756**
**Email: rbuskens@atl.lmco.com**

**www.sprucecommunity.org**

AFRL
THE AIR FORCE RESEARCH LABORATORY
LEAD | DISCOVER | DEVELOP | DELIVER

*Information Directorate*



*"The first essential of air power is pre-eminence in research"*

*- General H.H. Arnold, 1944*

**http://www.wpafb.af.mil/afrl/ri**


AFRL
THE AIR FORCE RESEARCH LABORATORY
LEAD | DISCOVER | DEVELOP | DELIVER